Lauren Miller
CS162
Final Project

Requirements

- Each room will be a class with up to four pointer variables that link to other rooms.
- There must be at least 10 rooms
- There will be a room superclass that will not be implemented.
- There will be at least 3 derived classes for rooms that have unique characteristics other just than a different name.
- There will be a goal for the player.
- The player will have a backpack, attaché, or knitting bag to carry items. The bag must have some limit.

Concept

The user plays a passenger on a sinking ship. The passenger can take items from rooms on the ship or talk to the other characters for ideas about what to take. However, every action takes time. The rooms fill with water, one by one, until the ship sinks. If the passenger has found a lifeboat, they survive. Then, depending on the direction the passenger goes, they may be stranded, picked up by pirates, washed ashore a deserted island, or be attacked by a giant squid. Each of these situations requires a different set of items to survive, without which the passenger will be sent to a watery grave.

Design

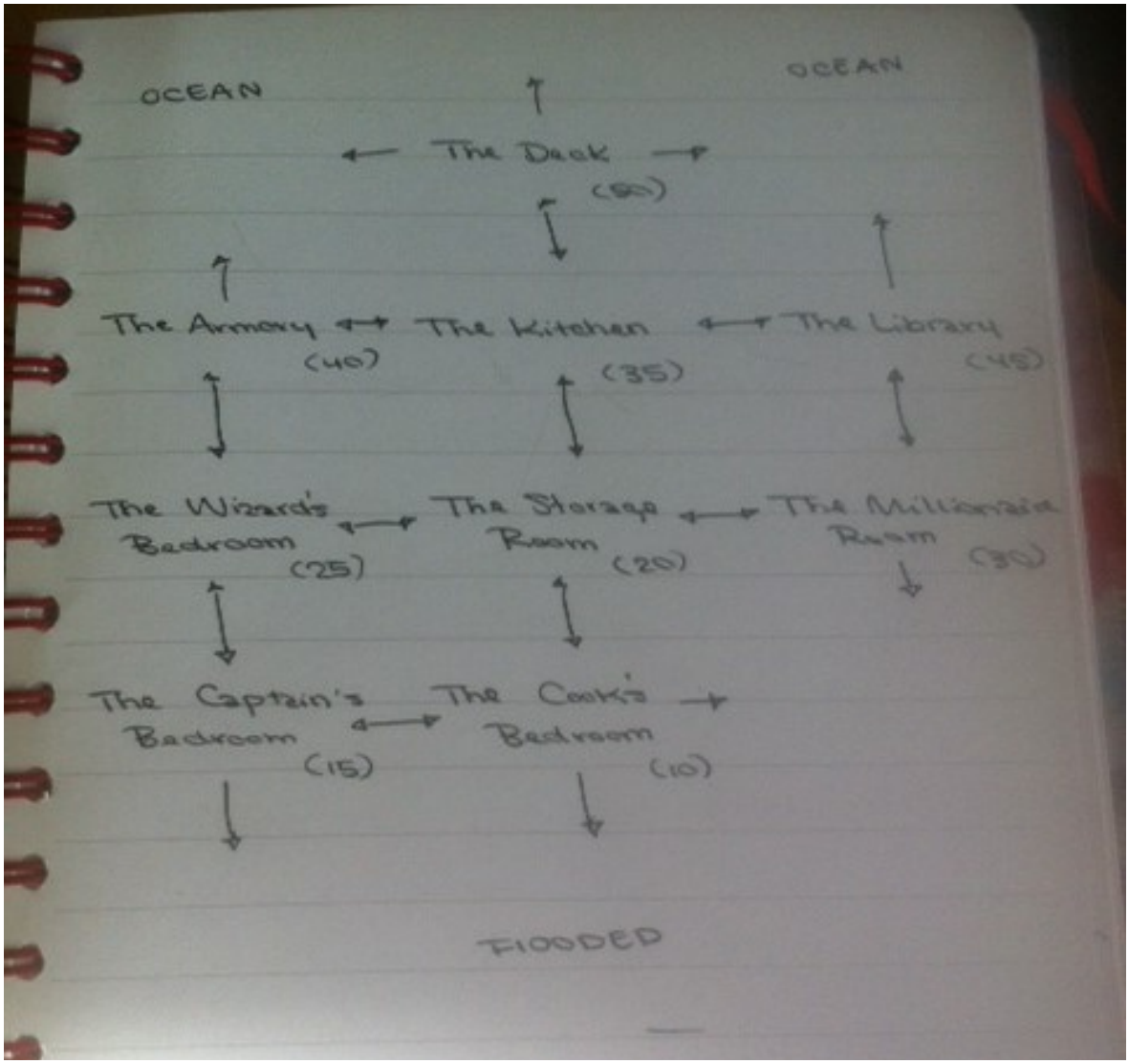
- There is, effectively, a time limit for the user to escape the sinking ship. Each time the user enters a room, it takes a minute. Talking to someone takes a minute. Putting an item in their bag takes a minute. Removing an item from the bag takes a minute. If the user is not on the deck, but in one of the rooms of the ship, when a room is flooded then the passenger dies
- The user has an array/bag that can hold five items. If the user tries to put an item in the bag and there is no space, the user is told. The user can also remove items from the bag.
- The Room superclass will have:
    - 4 pointers to other rooms or outside
    - a string that is the name of the room or “a flooded room”
    - a bool variable to say whether or not the room is filled with water
    - an int variable the determines at what minute the room is flooded
    - a string vector of items that the user can take
    - a virtual dostuff function that either tells the user what's in the room, talk to people there, and add items to their bag OR tell users their fate if they're in a final room -takes and returns time.
    - A picknextroom that lets user choose which way to go, virtual for final rooms, ocean, and closets
    - all the requisite set and get functions
- The subclasses of rooms are:
    - Rooms with another person in it
        - Includes string for other person

        - Will include a dostuff function that allows the user to “talk” to the person in the room
        - The deck
            - Floods at 50 minutes
            - One passengers, one who tells the user that there is a lifeboat on board, and who suggests that passengers may want to bring things onto the lifeboat
            - Has a buffet table, mashed potatoes, a mime, a Christmas tree, a pet cat, a bottle of wine, and a pile of magazines.
        - The kitchen
            - Floods at 35 minutes
            - The cook, who warns of pirates and says they can be bribed
            - Has a spatula, a blender, a carrot, a crate of eggs, a spoon, a bag of apples.
        - The storage room
            - Floods at 20 minutes
            - Another passenger, a millionaire, who mentions a deserted island
            - Has an inflatable life boat, a DVD player, a string of lights, snow shoes, a magic wand that sparkles, and a set of keys.
        - The library
            - Floods at 45 minutes
            - A wizard who says that he'll just turn himself into a fish and mentions a potion
            - Has books on survival, fancy knots, newts, lip gloss, dangerous sea creatures, evolutionary theory, and credit card fraud
        - The armory
            - Floods at 40 minutes
            - A crying captain who mentions a giant squid and that is likes apples
            - Has an axe, a pair of boxing gloves, a cannon ball, and and a box of gunpowder.
- The bedrooms:
    - Dostuff only talks about items
    - Have a mattress, bedspread, pillow, lamp, side table
    - The bedrooms are:
        - The wizard's, which has a potion and a hat and floods at 25 minutes
        - An millionaire's, which has a credit card and a sack full of cash and floods at 30 minutes
        - The cook's, which has a bag of apples and floods at 10 minutes
        - The captain's, which has a map and an egg and floods at 10 minutes
- The ocean

    - Dostuff only tells user the situation and what they need to survive
    - In picknextroom, if the user goes into the ocean without a lifeboat they instantly freeze and die. Otherwise, they can choose to go the normal directions, N to the squid, E to pirates, S to the deserted island, and W to stranded
- Final “rooms”:
    - Dostuff only describes the situation and what the user needs to survive and

checks whether the user has the necessary items
    - Picknextroom is empty, returns NULL
    - includes key string
    - The squid
        - If the user has a bag of apples they can bribe the squid to take them to land.
        - Otherwise it eats them
    - The pirates
        - If the user has cash they can bribe the pirates to take them to land
        - Otherwise the pirates kill them
    - The deserted island
        - If the user has a survival handbook then they can live on the island until they die
        - Otherwise they starve to death
    - Stranded
        - If the user has a magic potion, they can turn into a mermaid or merman and live forever
        - Otherwise they die from dehydration
- There will be a vector for all the rooms, to keep arguments tidy.
- There will be a set game function that sets the details of the rooms
- There will be a room by room function that passes along time as one of the arguments, also the bag array, and checks flooded status to confirm that the user has not drowned while in the room
- an function to take the bag array and a vector array that moves items from one to another and doesn't let more than five items in the array, etc.

The layout of the ship is sketched out in the image below. Please excuse how rough it is. The side edges of the paper mean that those rooms point to the ocean, as the user could always jump out of a window to reach it.



Testing

Test Case	**Driver Functions**	**Expected Outcome**	**Observed Outcome**
The user can navigate between rooms.	Main(); setrooms(nextroom); nextroom->dostuff(bag, time); nextroom->picknextroom(bag, time)	The user can go north south east and west from each room and is told they are in a different, expected room each time	As expected between rooms, but I'm told the ocean is a flooded room and I can't go there. Once I set the floodtime for the room, the problem is fixed.
The user is told the correct time upon entering each room	Main(); setrooms(nextroom); nextroom->dostuff(bag,	The user is told increasing times as they enter each new room on	Expected.

	time); nextroom->picknextroom(bag, time)	the ship	
The user can not go into a room once the time is beyond when the room should sink	Main(); setrooms(nextroom); nextroom->dostuff(bag, time); nextroom->picknextroom(bag, time)	The user is told to pick another room when they try to go into a room that has sunk	Expected.
The user is told that they have drowned if they are in a room when it's scheduled to sink	Main(); setrooms(nextroom); nextroom->dostuff(bag, time); nextroom->picknextroom(bag, time)	The user is told that they have drowned if they are in a room when it's scheduled to sink	Expected.
The user can “talk” to another passenger	Main(); setrooms(nextroom); nextroom->dostuff(bag, time)	The user is read a phrase.	Expected.
The items in the room are listed when the user chooses to have them listed.	Main(); setrooms(nextroom); nextroom->dostuff(bag, time)	The items set to be in a room are listed.	Expected.
The user can pick up items and put them in the bag	Main(); setrooms(nextroom); nextroom->dostuff(bag, time); nextroom->picknextroom(bag, time)	The user is told they have the lifeboat if they have already picked it up and go into the ocean.	Expected, but then I'm immediately told to go to another room. That problem is fixed once I delete a stray break;
The items disappear from the room once the user picks them up	Main(); setrooms(nextroom); nextroom->dostuff(bag, time);	If the user has picked up an item and then has the items in the room listed, that item they have picked up is not listed again.	Expected, but I'm read the phrase empty. That's fixed by only printing non-empty strings.
The user is told that they can't pick up more than five items when they try to do so	Main(); setrooms(nextroom); nextroom->dostuff(bag, time);	The user is told that they can't pick up more than five items when they try to do so	Expected.
Nextroom function will eventually return NULL and end the program once the character has died.	Main(); setrooms(nextroom); while (nextroom != NULL)... nextroom->picknextroom(bag, time);	The program will end once the user's character has died.	Expected.

The user will survive going into the ocean if they have picked up the lifeboat	Main(); setrooms(nextroom); nextroom->dostuff(bag, time); nextroom->picknextroom(bag, time)	The user is told they climb into the lifeboat if they have picked it up and can then choose to go to one of the final four rooms	Expected.
The user will survive the squid if they have picked up the apples	Main(); setrooms(nextroom); nextroom->dostuff(bag, time); nextroom->picknextroom(bag, time)	The user is told they survive if they have the apples and go to the squid room	Expected.
The user will survive the pirates if they have a bag of cash	Main(); setrooms(nextroom); nextroom->dostuff(bag, time); nextroom->picknextroom(bag, time)	The user is told they survive if they have the bag of cash and go to the pirate room	Expected.
The user will survive the deserted island if they have the book on wilderness survival	Main(); setrooms(nextroom); nextroom->dostuff(bag, time); nextroom->picknextroom(bag, time)	The user is told they survive if they have the book on wilderness survival and go to the deserted island room	Expected.
The user will survive being stranded if they have the magic potion	Main(); setrooms(nextroom); nextroom->dostuff(bag, time); nextroom->picknextroom(bag, time)	The user is told they survive if they have the magic potion and go to the stranded room	Expected.
The user will not survive the squid if they have not picked up the apples	Main(); setrooms(nextroom); nextroom->dostuff(bag, time); nextroom->picknextroom(bag, time)	The user is told that they do not survive because they don't have the apples	Expected.
The user will not survive the pirates if they do have not a bag of cash	Main(); setrooms(nextroom); nextroom->dostuff(bag, time); nextroom->picknextroom(bag, time)	The user is told that they do not survive because they don't have the bag of cash	Expected.
The user will not survive the deserted island if they do not	Main(); setrooms(nextroom); nextroom->dostuff(bag,	The user is told that they do not survive because they don't have the book	Expected.

have the book on wilderness survival	time); nextroom->picknextroom(bag, time)		
The user will not survive being stranded if they do not have the magic potion	Main(); setrooms(nextroom); nextroom->dostuff(bag, time); nextroom->picknextroom(bag, time)	The user is told that they do not survive because they don't have the potion	Expected.

Reflection

It felt like designing this thing took just as long as coding it out, mostly because the requirements were so loose. It was very tempting to try make a lot of variations on the classes I now have, but I eventually decided against making more work than was necessary when my grader and I would be the only ones to appreciate variety. Of course, writing out the design section above and planning things out took the greatest amount of time. Still, once I was done with that, the coding took very little time. There were a few small changes I made, but the largest one was deciding the make the finalroom class descend from the ocean class which in turn descends from the room superclass, both just to try using a granchild class for the first time and because both the ocean and finalroom classes used keys, livephrases, and diephrases. I had the worst time trying to figure out what was going on when I was trying to use a room pointer to point to a function that only existed in the ocean class, so I ended up copying everything to a room, the parent class. It was a useful lesson about polymorphism, but it was extremely confusing at the time. One other important aspect that I changed was not returning a vector of the rooms from the set rooms function, but returning only one pointer to a room from the setrooms function. I realized that all the rooms were pointed to by other rooms, so it was very much like an array in that you just needed to find the start of it. It was a nice reminder that I've improved since taking this class, since I didn't really understand pointers very well at the start of it. Of course, later on I had the problem with pointing to descendant class functions, but I still feel much more confident in using pointers overall.